Package ‘sdols’

October 2, 2019

Type Package

Title Summarizing Distributions of Latent Structures

Version 2.0.0

URL https://dahl.byu.edu

BugReports https://dahl.byu.edu

Description Summaries of distributions on clusterings and feature allocations are provided. Specifically, point estimates are obtained by the sequentially-allocated latent structure optimization (SALSO) algorithm to minimize squared error loss, absolute error loss, Binder loss, or the lower bound of the variation of information loss. Clustering uncertainty can be assessed with the confidence calculations and the associated plot.

Imports salso (>= 0.1.3)

Depends R (>= 3.3.0)

LazyData TRUE

License Apache License 2.0 | file LICENSE

Encoding UTF-8

RoxygenNote 6.1.1

NeedsCompilation no

Author David B. Dahl [aut, cre],
Peter Müller [aut]

Maintainer David B. Dahl <dahl@stat.byu.edu>

Repository CRAN

Date/Publication 2019-10-02 16:40:06 UTC

R topics documented:

confidence *Compute Clustering Confidence*

Description

This function computes the confidence values for n observations based on a clustering estimate and the expected pairwise allocation matrix.

Usage

```
confidence(estimate, expectedPairwiseAllocationMatrix)
```

Arguments

`estimate` A vector of length n, where i and j are in the same cluster if and only if clustering[i] == clustering[j].

`expectedPairwiseAllocationMatrix` A n-by-n symmetric matrix whose (i,j) elements gives the estimated expected probability that items i and j are in the same cluster.

Author(s)

David B. Dahl <dahl@stat.byu.edu>

See Also

expectedPairwiseAllocationMatrix, dlso, salso

Examples

```
suppressWarnings({  # For testing purposes, suppress deprecation warning.

probabilities <- expectedPairwiseAllocationMatrix(iris.clusterings)
clustering <- salso(probabilities)
conf <- confidence(clustering, probabilities)
conf

})
```

dlso *Perform Draws-Based Latent Structure Optimization*

Description

Among the supplied latent structures, this function picks the structure that minimizes one of various loss functions.

Usage

```
dlso(x, loss = c("squaredError", "absoluteError", "binder",
  "lowerBoundVariationOfInformation")[1], multicore = TRUE,
  expectedPairwiseAllocationMatrix = NULL)
```

Arguments

`x` A collection of clusterings as a `B`-by-`n` matrix, each of the `B` rows represents a clustering of `n` items using cluster labels. For clustering `b`, items `i` and `j` are in the same cluster if `x[b,i] == x[b,j]`.

`loss` One of `"squaredError"`, `"absoluteError"`, `"binder"`, or `"lowerBoundVariationOfInformation"` to indicate the optimization should seeks to minimize expectation of the squared error loss, absolute error loss, Binder loss (Binder 1978), or the lower bound of the variation of information loss (Wade & Ghahramani 2017), respectively. The first three are equivalent.

`multicore` Logical indicating whether computations should take advantage of multiple CPU cores.

`expectedPairwiseAllocationMatrix`
A `n`-by-`n` symmetric matrix whose `(i,j)` elements gives the estimated expected number of times that items `i` and `j` are in the same subset (i.e., cluster). If `NULL`, it is computed from `x`.

Value

A list A clustering (as a vector of cluster labels).

Author(s)

David B. Dahl <`dahl@stat.byu.edu`>

References

Wade, S. and Ghahramani, Z. (2017). Bayesian cluster analysis: Point estimation and credible balls. Bayesian analysis.

Binder, D. (1978). Bayesian Cluster Analysis. Biometrika, 65: 31–38.

See Also

expectedPairwiseAllocationMatrix, salso

Examples

```
suppressWarnings({  # For testing purposes, suppress deprecation warning.

dlso(iris.clusterings)

})
```

---

expectedPairwiseAllocationMatrix

Compute Expected Pairwise Allocation Matrix

Description

This function computes the n-by-n matrix whose (i,j) element gives the (estimated) expected number of times that i and j are in the same subset (i.e, cluster). This is the (estimated) probability that items are clustered together. These estimates are based on the frequencies from the supplied, randomly-sampled clusterings.

Usage

```
expectedPairwiseAllocationMatrix(x)
```

Arguments

x	A collection of clusterings as a B-by-n matrix, each of the B rows represents a clustering of n items using cluster labels. For clustering b, items i and j are in the same cluster if x[b,i] == x[b,j].

Value

A n-by-n symmetric matrix whose (i,j) elements gives the estimated expected number of times that items i and j are in the same subset (i.e, cluster) based on the frequencies from the supplied clusterings.

Author(s)

David B. Dahl <dahl@stat.byu.edu>

See Also

dlso, salso

Examples

```
suppressWarnings({  # For testing purposes, suppress deprecation warning.

probabilities <- expectedPairwiseAllocationMatrix(iris.clusterings)

})
```

`iris.clusterings` *Clusterings of the Iris Data*

Description

Randomly generated clusterings of the iris dataset.

Usage

```
iris.clusterings
```

Format

A 1000-by-150 matrix of 1000 randomly generated clusterings of the 150 observations in the iris dataset.

See Also

`iris`

`latentStructureFit` *Compute Fit Summaries for a Latent Structure Estimate*

Description

This function computes various summaries of the fit of a clustering based on the expected pairwise allocation matrix.

Usage

```
latentStructureFit(estimate, expectedPairwiseAllocationMatrix)
```

Arguments

`estimate` A clustering. If `estimate` is a length `n` vector, it is taken to be a clustering where items `i` and `j` are in the same cluster if and only if `estimate[i] == estimate[j]`.

`expectedPairwiseAllocationMatrix`
A `n`-by-`n` symmetric matrix whose `(i,j)` elements gives the estimated expected number of times that items `i` and `j` are in the same subset (i.e., cluster).

Value

A list of the following elements:

absoluteError The expectation of the absolute error loss.

binder The expectation of the binder loss.

lowerBoundVariationOfInformation The lower bound of the expectation of the variation of information loss.

Author(s)

David B. Dahl <dahl@stat.byu.edu>

See Also

expectedPairwiseAllocationMatrix, salso

Examples

```
suppressWarnings({  # For testing purposes, suppress deprecation warning.

probabilities <- expectedPairwiseAllocationMatrix(iris.clusterings)
estimate <- salso(probabilities)
latentStructureFit(estimate, probabilities)

})
```

salso *Perform Sequentially-Allocated Latent Structure Optimization*

Description

This function implements the sequentially-allocated latent structure optimization (SALSO) to find a clustering that minimizes various loss functions. The SALSO method was presented at the workshop "Bayesian Nonparametric Inference: Dependence Structures and their Applications" in Oaxaca, Mexico on December 6, 2017.

Usage

```
salso(expectedPairwiseAllocationMatrix, loss = c("squaredError",
  "absoluteError", "binder", "lowerBoundVariationOfInformation")[1],
  nCandidates = 100, budgetInSeconds = 10, maxSize = 0,
  maxScans = 10, multicore = TRUE)
```

Arguments

`expectedPairwiseAllocationMatrix`
: A `n`-by-`n` symmetric matrix whose `(i,j)` elements gives the estimated expected number of times that items `i` and `j` are in the same subset (i.e., cluster).

`loss`
: One of `"squaredError"`, `"absoluteError"`, `"binder"`, or `"lowerBoundVariationOfInformation"` to indicate the optimization should seeks to minimize squared error loss, absolute error loss, Binder loss (Binder 1978), or the lower bound of the variation of information loss (Wade & Ghahramani 2017), respectively. The first three are equivalent.

`nCandidates`
: The (maximum) number of candidates to consider. Fewer than `nCandidates` may be considered if the time in `budgetInSeconds` is exceeded. The computational cost is linear in the number of candidates and there are rapidly diminishing returns to more candidates.

`budgetInSeconds`
: The (maximum) number of seconds to devote to the optimization. When this time is exceeded, no more candidates are considered.

`maxSize`
: Either zero or a positive integer. If a positive integer, the optimization is constrained to produce solutions whose number of clusters is no more than the supplied value. If zero, the size is not constrained.

`maxScans`
: The maximum number of reallocation scans after the intial allocation. The actual number of scans may be less than `maxScans` since the algorithm stops if the result does not change between scans.

`multicore`
: Logical indicating whether computations should take advantage of multiple CPU cores.

Value

A clustering (as a vector of cluster labels).

Author(s)

David B. Dahl <dahl@stat.byu.edu>

References

Wade, S. and Ghahramani, Z. (2017). Bayesian cluster analysis: Point estimation and credible balls. Bayesian analysis.

Binder, D. (1978). Bayesian Cluster Analysis. Biometrika, 65: 31–38.

See Also

`expectedPairwiseAllocationMatrix`, `dlso`

Examples

```
suppressWarnings({  # For testing purposes, suppress deprecation warning.

probabilities <- expectedPairwiseAllocationMatrix(iris.clusterings)
salso(probabilities)

})
```

Index